Negotiation Strategies Considering Opportunity Functions for Grid Scheduling

Jiadao Li[1], Kwang Mong Sim[2], and Ramin Yahyapour[1]

[1] Members of the Core Grid Institute on Resource Management and Scheduling
Institute for Robotics Research - Information Technologies
University of Dortmund, 44221 Dortmund, Germany
jiadao.li@uni-dortmund.edu,
ramin.yahyapour@uni-dortmund.edu
[2] Department of Computer Science, Hong Kong Baptist University
Kowloon Tong, KLN, Hong Kong
prof_sim_2002@yahoo.com

Abstract. In Grid systems, nontrivial qualities of service have to be provided to users by the resource providers. However, resource management in a decentralized infrastructure is a complex task as it has to cope with different policies and objectives of the different parties: providers and consumers/users. Agreement-based resource management is considered to solve many of these problems as the conflicts between the users and resource providers can be reconciled in a negotiation process. Such negotiation processes must be automated with no or minimal human interaction, considering the potential scale of Grid systems and the amount of necessary transactions. Therefore, strategic negotiation models and strategic negotiation strategies play important roles. In this paper, negotiation strategies considering time and opportunity functions for Grid scheduling are proposed and examined. The simulation results demonstrate that the negotiation strategies are suitable and effective for Grid environments.

1 Introduction

Grid computing [1,2] is a Service Oriented Architecture (SOA [3,2]) in which a resource user typically requires a certain service quality to be provided by the resource owners. Here, agreement based resource management [4] is typically considered as a suitable approach for this scenario. Negotiation is the process towards creating suitable agreements between different parties in a Grid. The negotiation process in a Grid computing environment should be done automatically and transparently with the growing scale in Grids [5]. In order to automate the negotiation process, suitable negotiation models are required that take the different policies and objectives of the resource providers and resource users into account and produce suitable service level agreements in reasonable time with minimized or even no user and provider interference.

In our previous work [6,7], a strategic negotiation model which supports the automatic negotiation in Grid computing was proposed and evaluated. In that model, the user, or more precisely some meta/grid-scheduling agent or job broker on his behalf, will contact different resource providers, negotiate with several of them and make a

A.-M. Kermarrec, L. Bougé, and T. Priol (Eds.): Euro-Par 2007, LNCS 4641, pp. 447–456, 2007.

decision to commit to a particular agreement with one resource provider. This is considered as the *one to many* negotiation type [8]. A *Concurrent bilateral negotiation model* [9] is suitable for this problem. This paper is an extension of the previous work. In this paper, the strategic negotiation model proposed in [6] is adopted, while we now add negotiation strategies considering the opportunity functions. An opportunity function determines the bargaining position of a negotiation agent based on available outside options [10]. A Grid resource management system needs to continuously adapt to changes in the availability of computing resources (i.e., outside options). These strategies have been implemented and evaluated.

2 Related Work

There are many approaches proposed for the Grid resource management problem, for example, economic methods. An overview of such methods can be found in [11] by Buyya et al., or in in [12] by Ernemann et al. , or by Wolski et al. in [13], or by Lai et al. in [14,15]. In these papers, economic based resource management in Grid computing are investigated and several economic models are evaluated. To this end, a lot of effort has been made on Grid resource management with support of service level agreements (SLAs). In [4], the concepts of agreement-based resource management in the Grid computing environment are introduced and a general agreement model is presented; in [16], a Grid resource broker supporting SLAs called GRUBER is presented and evaluated in a real grid. In [17], the very few existing research initiatives on applying bargaining as a mechanism for managing Grid resources are reviewed and compared.

In this paper, we now adopt the negotiation strategies considering the time and opportunity functions [10] in the earlier proposed negotiation model for Grid scheduling and present an evaluation with discrete event-based simulation.

3 Negotiation Model

As a bilateral negotiation model is the building block of concurrent negotiation model, we will briefly introduce this first in this section.

3.1 Bilateral Negotiation Model

There are three parts in the bilateral negotiation model [18]: 1) the negotiation protocol, 2) the used utility functions or preference relationships for the negotiating parties, and 3) the negotiation strategy that is applied during the negotiation process. In this paper, we adopted Rubinstein's sequential alternating offer protocol in Grids, see [19]. In the negotiation process, when one negotiation side times out or an agreement is created, the negotiation process will end. Disagreement is considered as the worst outcome, therefore, the negotiation party will always try to avoid opting out of the negotiation. In this negotiation model, the negotiation parties do not know the opponents' private reservation information and their preferences/utility functions.

Typically, the objectives of a user for a computational job are to obtain a shorter response time and/or to get cheaper resources, while the resource providers expect to gain higher profit and/or higher utilization. However, our model is not restricted to particular objectives and can be flexibly defined for different scenarios. In real Grid systems, there can be many different negotiation objectives, which are interdependent and should be simultaneously dealt with which yields to a multi-criteria optimization problem [20]. In this presented research work, without limiting generality we restrict our analysis on considering the expected waiting time of the job and the expected cost per cpu time. However, the model can be applied and extended to other criteria as well. For the resource providers, we also assume two corresponding negotiation issues: the expected waiting time until a job can be started $T_s^t(Job)$, and the expected price $P_s^t(Job)$. The expected waiting time for the newly incoming job can be obtained from the current resource status and the future schedule plan considering other created agreements which have to be fulfilled. The expected price will be obtained via the negotiation process. In paper [6], more details about the utility functions are given.

3.2 Negotiation Strategies Considering the Time and Opportunity Functions

In the negotiation process, it is assumed that both of the negotiation agents behave according to *the good-faith bargaining principles* which means that it is usually not easily reversed [21]. Here, on the basis of the initial offer values, successive offers by sellers are monotonically decreasing while successive offers by the buyers are monotonically increasing. In order to create the agreement, both of the negotiation parties want to narrow the difference between the offers and counter offers with respect to different negotiation issues. In the strategic negotiation model, the negotiation agents can take different kinds of negotiation strategies developed in the agent community [22] to create the negotiation offer at different negotiation times. In [10], the negotiation strategies are proposed and analyzed for market-driven agents to make prudent compromises taking into account factors such as time preference, opportunity functions, competition factors.

In a multilateral negotiation, having outside options may give a negotiator more bargaining "power". However, negotiations may still break down if the proposals between two negotiators are too far apart.

Suppose agent B (the job agent) engages S_j (the resource provider) in round t. At any negotiation round t, $B's$ last proposal (bid) is represented by a utility vector $(V_t^{B\rightarrow S_j}, W_t^{B\rightarrow S_j})$ and $S_j's$ proposal (offer) is a utility vector $(V_t^{S_j\rightarrow B}, W_t^{S_j\rightarrow B})$.

The opportunity function

$$O(n_t^B, v_t^{B\rightarrow S_j}, \langle W_t^{S_j\rightarrow B}\rangle) = 1 - \prod_{j=1}^{n_t^B} \frac{V_t^{B\rightarrow S_j} - W_t^{S_j\rightarrow B}}{(V_t^{B\rightarrow S_j} - c^B)} \tag{1}$$

determines the amount of concession based on 1) trading alternatives (number of trading parties n_t^B) and differences in utilities $V_t^{B\rightarrow S_j}$ generated by the proposal of the job agent and the counter-proposal of its trading party $W_t^{S_j\rightarrow B}$. c^B is the worst possible utility for agent B. Space limitation precludes detailed derivations from being included here, but they can be found in [10].

As explained before, the negotiation party will also modify the negotiation offer with the negotiation time going on. There are many ways of defining the function $\alpha_j^a(t)$ to model the effects of the remaining negotiation time. We also use the following function to calculate the $\alpha_j^a(t)$, see [6]:

$$\alpha_j^a(t) = k_j^a + (1 - k_j^a)(\frac{t}{t_{max}^a})^{1/\beta}, \tag{2}$$

where t_{max}^a is the deadline of the negotiation party a for the completion of the negotiation, t denotes the current time instant in the available negotiation time set, the parameter β is the degree of convexity that determines the type of the negotiation party in the time dependent strategy. Different β values yield different negotiation strategies. For the initial bargaining value k_j^a is used, for which the following relation holds $0 \leq k_j^a \leq 1$.

As pointed out in [10], there are several means of combining the time and the opportunity function effects to create the offers for the negotiation parties, for instance, $0.5 * (T(t) + O(t))$, or $T(t) * O(t)$. Here we use the former one.

It is assumed that P_c^t is the offered price at time t by the user, P_s^t is the offered price at time t by the resource provider; $T_c^t(job)$ is the proposed waiting time at time t by the user, $T_s^t(job)$ is the acceptable waiting time for the specific job at time t according to the current resource status considering the future reserved resource as well.

We assume that V_j is the utility function of the negotiation party which associates with the negotiation issue j and the $x_{a\rightarrow b}^t[t]$ is the offer provided by one party (denoted by a) to another negotiation party (denoted by b).

If V_j is decreasing:

$$x_{a\rightarrow b}^t[t] = a_j^t + 0.5((min(max_j^a, b_j^t) - a_j^a) * (O(t) + \alpha_j^a(t)), \tag{3}$$

if V_j is increasing:

$$x_{a\rightarrow b}^t[t] = a_j^t + (1 - 0.5 * (\alpha_j^a(t) + O(t))(min(max_j^a, b_j^t) - a_j^t), \tag{4}$$

Equations 3 and 4 represents the job user's strategy and the resource provider's strategy respectively.

As there are two negotiation issues involved in this negotiation process, we assume that if the offer in which one of the negotiation issues from the opponent is satisfied, then it will accept this value and not further change it but only change the value of the remaining other issue in the following negotiation process. This is just a first heuristic to analyze the behavior of the negotiation strategies. In real life, negotiation issues will not be independent and thus acceptable deal is not easily reached. This will require more complex negotiation strategies that need to be considered in future work. For now, we accept that the negotiation issues are modified according to the previously made assumption on the monotonous increase/decreas by the parties.

3.3 Concurrent Bilateral Negotiation Model

As mentioned above, in the Grid environment a number of resources will typically be available which are capable of fulfilling the job constraints after the resource discovery phase. The user or a corresponding scheduling component will contact different resource providers and initiate the negotiation process for the actual resource allocation. The negotiation relationship is of the "one to many" type, which can be treated as a concurrent bilateral model. In the concurrent negotiation threads in which a single user is involved, the reservation value of the negotiation issues and preferences are the same. However, the user may adopt different strategies with respect to different negotiation opponents. Furthermore, they might change the negotiation strategies during the negotiation process based on available information from different negotiation threads.

Because these negotiation threads are executed concurrently, it is very difficult to predict whether the user might achieve a better offer from another negotiation thread if there is already a suitable offer that could be committed to an agreement. In our model, we assumed that once an agreement is available, it will be created and committed. Of course, in a real life scenario the job agent might actually exploit the available time to find several offers and decide at the end on the best offer. In paper [23], we analyze the results of tradeoff between the “best” and the “first available” agreement. In this paper, for simplicity, we restricted our examination to accepting the first available agreement. If one negotiation thread is successfully negotiated, all of the other negotiation threads will be terminated. The agreement can then be used by provisioning and execution service to actually start a job on the local resource management system.

4 Evaluation

Discrete event simulation has been used to evaluate the proposed negotiation model. Currently, there is no real data from Grid computing environments that include suitable information for negotiation models. However, high performance computing is still the typical application scenario for Grid technology. For this scenario, workload traces are available which were recorded on actual machine installations [24]. Therefore, our first evaluations are based on such traces. However, negotiation information are not included in this data as none of the real systems supported negotiation models yet. To this end, the missing information can only be modeled based on first assumptions. In the following the simulation configuration is described and the simulation results are analyzed.

4.1 Simulation Configuration

At the beginning of the negotiation, the negotiation parties will always make the offers which are most favorable to themselves. So we assume initial values of 0 for k_j^a of all the negotiation parties. For performance analysis we assume a negotiation interval of 1s between each negotiation round. In the following we describe the models of the users and the resource providers. In order to evaluate the learning-based negotiation algorithms, we will compare simulation results in different simulation cases.

User Model. In our simulation we consider parallel batch jobs in an online scenario. Typically, users will behave quite differently in the negotiation process. For our simulation, we assume two different kinds of user objectives: time-optimization and cost-optimization. The actual behavior of real users will be investigated in future research work. Below are the parameters of the user modeling which have been applied for the simulation.

- Negotiation span is uniformly distributed in $[0, 30]$s.
- Maximum price of the different job user is uniformly distributed in $[4.0, 9.0]$.
- Acceptable waiting time for the job users are uniformly distributed in $[0, 36000]$s.
- For the tough negotiator, β value is uniformly distributed in $[0.02, 0.2]$.
- For the conceder negotiator, β value is uniformly distributed in $[20, 40]$.
- Weights of time and price for the time-optimization are 0.8 and 0.2, while the weights of the time and price for the cost-optimization are 0.2 and 0.8.

Resource Provider. For the local resource management system an FCFS scheduling strategy with backfilling [25] is adopted which is common for parallel computers. There is no preemption allowed in our scenario. To this end, in this evaluation we do not consider the co-allocation of resources from different providers. The resources are all homogeneous and only differ in the number of available CPU nodes at each site. The simulated hardware configurations of the resource providers are consistent with actual configurations of the systems from which the real traces are originated. In this paper, we present results for traces from the Cornell Theory Center [24] which had 512 CPU nodes. In our simulation we assumed a Grid scenario with 6 different machines (parallel computer or cluster with a given set of CPU nodes) and therefore 6 resource providers. However, to stay consistent with the available workload from the CTC traces, the total number of nodes for all simulated machines is again 512 nodes. The number of nodes on each machine and the negotiation parameters for each resource provider are given below.

- The numbers of CPU nodes for the machines are {384,64,16,16,16,16}.
- Their different maximum prices per CPU time are {8.2, 8.0, 7.5, 7.6, 7.4, 7.5}.
- Their different minimum prices per CPU time are {2.4, 2.3, 2.0, 1.95, 1.90, 1.80}.
- Negotiation deadlines of different resource providers are all 30s, which means that usually the resource provider will not opt out of the negotiation once the negotiation thread is created.
- For the conceding negotiator, β value is {32, 35, 34, 38, 40, 40}.
- For the tough negotiator, β value is {0.03,0.05,0.04,0.10,0.05,0.06}

4.2 Evaluation Criteria

In the following we provide some first evaluation remarks which give some qualitative information about the performance of the model. The actual quality will have to be verified with better workload models and real implementations.

- Comparison between the negotiation result and the reference point [21], which is the middle of the zone of possible agreement of user and resource provider: $[C_j^{max}, S_j^{min}]$. The reference point is computed by the following function:

$$U_j^{ref} = \frac{C_j^{max} + S_j^{min}}{2} \tag{5}$$

- The rate of successfully created agreement for all jobs.
- The negotiation overhead to create the agreement measured by the time taken to create the agreement. In our case, we use the final negotiation rounds which represents the required number of messages exchanged. The actual network overhead will depend on the actual network speed for this message exchange.
- In computational Grids, the users will concern about the response time and waiting time of a job, while for the resource providers the utilization and the profit will probably be the main objectives. We also compare these criteria to get some feedback about the feasibility of the negotiation model.

Simulation Results. We used the first 10000 jobs from the CTC workload traces [24] to conduct our simulation. We compare the on average required number of negotiation rounds for the successful creation of an agreement, and the rate of successfully created agreements in comparison to the total number of job requests. Other criteria are the average weighted response time (AWRT), the average weighted waiting time (AWWT), the average price difference between the agreement price (AP) and the reference price (RP). For the weight in AWRT and AWWT, the job resource consumption is used, see [26]. This weight prevents any favor of jobs with high or low resource consumption over each other.

In order to evaluate the simulation results, we compared the simulation cases with the associated simulation cases that we did in previous publications [6]. The following simulation cases are considered. Case 1: Both of them use the conceding strategy [6]; Case 2: Both of them use the conceding strategy and opportunity functions; Case 3: Both of them use the linear strategy [6]; Case 4: Both of them use linear strategy and opportunity functions; Case 5: Both of them use the tough strategy [6]; Case 6: Both of them use tough strategy and opportunity functions;

The success rate of negotiations in these 6 cases are as the following tables. The successful rate of negotiations in these 10 cases are 94.65%, 94.03%, 62.87%, 53.49%, 1.95%, 43.40% respectively, so we can see that using the time and opportunity together can yield higher creation rate in Case 6 than Case 5; in other simulation cases the creation rate are comparable.

Figure 1 shows a selection of results. R1 to R6 stands for the resources from 1 to resource 6 respectively. From these simulation results, we can see that the negotiation agents using the time and opportunist functions to narrow the differences between the offers and counter offers can achieve higher utilities than using negotiation strategies in our former work in [6]. In the simulation cases which use the time and opportunity functions, the AWWT is less than the cases in [6]. Except in the Case 5, there are no agreements created in R1, so the AWWT is 0, in other resources (except R2 and R4), the jobs can be started immediately due to the quite lower utilization rate. But the users usually pay more for the needed resources than in the simulation cases we did before as shown in the result figure. AWRT is comparable and in the same range as for Grid models which do not use negotiation models but conventional queuing systems. That means, the presented model can be considered feasible for real Grid infrastructure as it does not lead to any drawbacks in the performance results. To the contrary the negotiated waiting time of the jobs will be guaranteed by the resource providers which is the anticipated quality of service level and can be seen as a major asset of

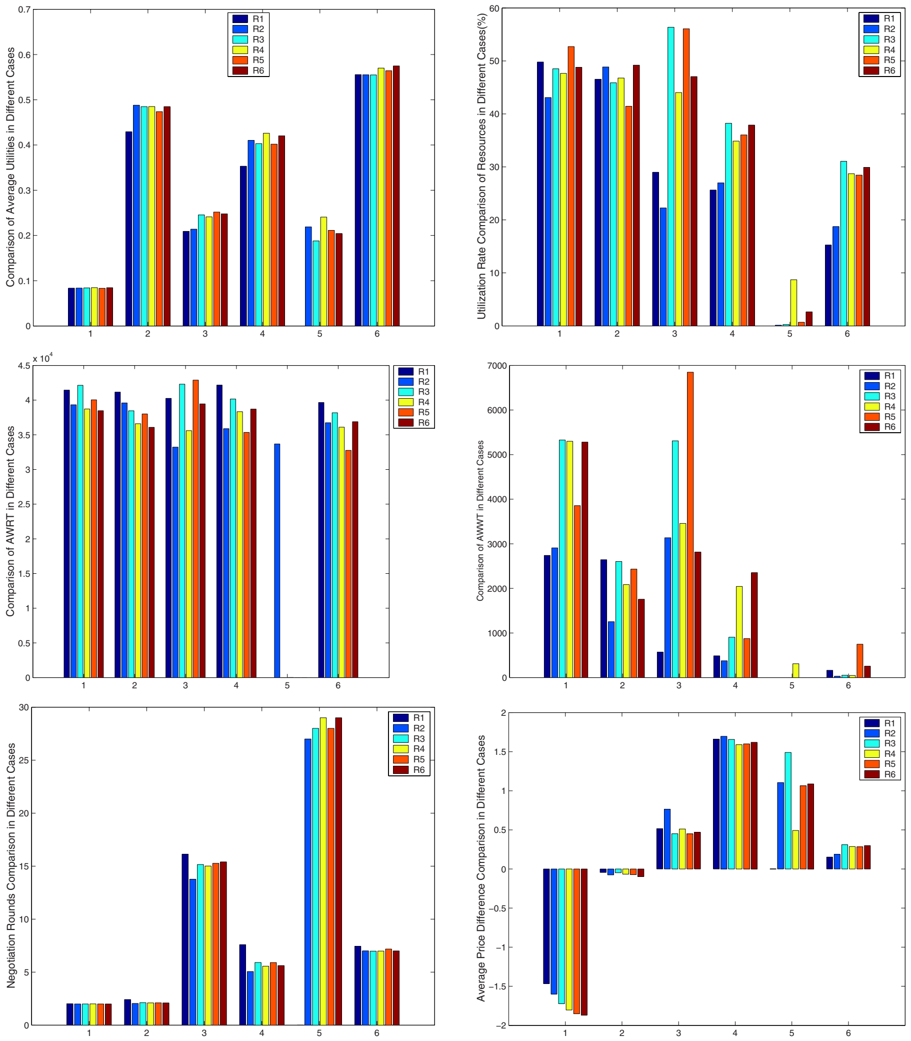

Fig. 1. Comparison between different negotiation cases including the results for the individual six resources

such an approach. Also we can see that the simulation cases using the opportunity functions can work with less negotiation rounds than using the pure negotiation tactic. An agent using opportunity function is more likely to reach a quicker agreement because it has higher chance of exploring more negotiation options. From these simulations, we can see that negotiation strategies considering time and opportunity functions are quite flexible and effective, and can actually be used in the dynamically changing Grid infrastructure.

5 Conclusions and Future Work

In this paper, we proposed and evaluated a strategic negotiation model in the Grid scenario. This model has been evaluated using discrete event based simulation. The results show that it can be applied in the practical use in automatic job scheduling. The presented results can be seen as first steps in analyzing the features and requirements for automatic negotiation strategies. However, the actual evaluation of the obtained service quality is difficult to obtain as there is no valid user job and preference workload model for Grids available which takes economic functionality into account. The presented results indicate that the negotiation overhead in terms of exchanged messages is manageable for practical application. In the future work, we will try other combinations of opportunity and time functions which can be used as the alternatives for the negotiation strategies for Grid scheduling.

Acknowledgement

This research work is partially carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265). K. M. Sim acknowledges financial support from the Hong Kong Research Grant Council (project code: RGC/HKBU210906). The authors would like to thank the anonymous referees for the comments and suggestions.

References

1. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: Enabling scalable virtual organizations. In: Sakellariou, R., Keane, J.A., Gurd, J.R., Freeman, L. (eds.) Euro-Par 2001. LNCS, vol. 2150, Springer, Heidelberg (2001)
2. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (2003)
3. The w3c web services architecture working wroup public draft (February 2004), `http://www.w3.org/TR/ws-arch/`
4. Czajkowski, K., Foster, I., Kesselman, C.: Agreement-based resource management. Proceedings of the IEEE 93(3), 631–643 (2005)
5. Foster, I., Jennings, N.R., Kesselman, C.: Brain meets brawn: Why grid and agents need each other. In: Kudenko, D., Kazakov, D., Alonso, E. (eds.) AAMAS II, 2005. LNCS (LNAI), vol. 3394, pp. 8–15. Springer, Heidelberg (2005)
6. Li, J., Yahyapour, R.: Negotiation strategies for grid scheduling. In: GPC 2006. LNCS, vol. 3947, pp. 42–52. Springer, Heidelberg (2006)
7. Li, J., Yahyapour, R.: Learning-based negotiation strategies for grid scheduling. In: Proceedings of the International Symposium on Cluster Computing and the Grid (CCGRID2006), pp. 567–583. IEEE Computer Society Press, Los Alamitos (2006)
8. Lomuscio, A.R., Wooldridge, M., Jennings, N.R.: A classification scheme for negotiation in electronic commerce. International Journal of Group Decision and Negotiation 12(1), 31–56 (2003)
9. Nguyen, T.D., Jennings, N.R.: A heuristic model of concurrent bi-lateral negotiations in incomplete information settings. In: Proc. 18th Int. Joint Conf. on AI, Acapulco, Mexico, pp. 1467–1469. IEEE Computer Society Press, Los Alamitos (2003)

10. Sim, K.M.: Equilibria, prudent compromises, and the ”waiting” game. IEEE Transactions on Systems, Man, and Cybernetics, Part B 35(4), 712–724 (2005)
11. Buyya, R.: Economic-based Distributed Resource Management and Scheduling for Grid Computing. PhD thesis, Monash University, Melbourne, Australia (2002)
12. Ernemann, C., Yahyapour, R.: Applying Economic Scheduling Methods to Grid Environments. In: Grid Resource Management - State of the Art and Future Trends, pp. 491–506. Kluwer Academic Publishers, Dordrecht (2003)
13. Wolski, R., Plank, J.S., Brevik, J., Bryan, T.: Analyzing market-based resource allocation strategies for the computational grid. International Journal of High Performance Computing Applications 15(3) 258–281 (2001)
14. Lai, K., Rasmusson, L., Adar, E., Sorkin, S., Zhang, L., Huberman, B.A.: Tycoon: an Implemention of a Distributed Market-Based Resource Allocation System. Technical Report arXiv:cs.DC/0412038, HP Labs, Palo Alto, CA, USA (December 2004)
15. Kevin Lai, B.A.H., Fine, L.: Tycoon: A Distributed Market-based Resource Allocation System. Technical Report arXiv:cs.DC/0404013, HP Labs, Palo Alto, CA, USA (April 2004)
16. Dumitrescu, C., Foster, I.T.: Gruber: A grid resource usage sla broker. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005. LNCS, vol. 3648, pp. 465–474. Springer, Heidelberg (2005)
17. Sim, K.M.: A survey of bargaining models for grid resource allocation. ACM SIGecom Exch. 5(5), 22–32 (2006)
18. Kraus, S.: Strategic Negotiation in Multi-Agent Environments. MIT Press, Cambridge (2001)
19. Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica 50, 97–110 (1982)
20. Kurowski, K., Nabrzyski, J., Oleksiak, A., Weglarz, J.: Multicriteria Aspects of Grid Resource Management. In: Grid Resource Management - State of the Art and Future Trends, pp. 271–295. Kluwer Academic Publishers, Dordrecht (2003)
21. Raiffa, H.: The Art and Science of Negotiation. Harvard Universtiy Press, Cambridge (1982)
22. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: Prospects, methods and challenges. Int. J. of Group Decision and Negotiation 2(10), 199–215 (2001)
23. Li, J., Yahyapour, R.: A strategic negotiation model for grid scheduling. International Transactions on Systems Science and Applications 1(4), 411–421 (2006)
24. Standard workload format (July 2005), `http://www.cs.huji.ac.il/labs-parallel/workload/index.html`
25. Lifka, D.: The ANL/IBM SP scheduling system. In: Feitelson, D.G., Rudolph, L. (eds.) IPPS 1995. LNCS, vol. 949, pp. 295–303. Springer, Heidelberg (1995)
26. Schwiegelshohn, U., Yahyapour, R.: Analysis of first-come-first- serve parallel job scheduling. In: Proceedings of the 9th SIAM Symposium on Discrete Algorithms, pp. 629–638 (January 1998)